

FEATURE SELECTION AND CLASSIFICATION FOR MICROARRAY DATA USING UPGRADE CHI-SQUARE TEST

¹Lwin May Thant

Faculty of Computer Science,
University of Computer Studies, Yangon,
Myanmar,

lwinmaythant@ucsy.edu.mm, lwinmaythant333@gmail.com

²Tin Zar Thaw

Faculty of Computer Science,
University of Computer Studies, Yangon,
Myanmar,

tinzarthaw@ucsy.edu.mm

Abstract

Microarray technology allows the monitoring of thousands of gene expressions in various biological contexts. The purpose of this paper is to lay the groundwork for the proposal and creation of a new algorithm based on unbalanced classes that will upgrade the original Chi-square algorithm. The proposed UpgCHI over the Apache Spark framework based on unbalanced classes likely represents their efforts to provide a more robust and reliable solution in comparison to the original Chi-square method to handle multi-class problems. The results are compared with the original Chi-square test and an upgraded UpgCHI multiclass selection algorithm on microarray datasets using different three classifiers to evaluate the performance. The presented method generates good classification performance with an UpgCHI test, which demonstrates an increased accuracy of 96% in DLBCL 2Classes, 89% in Colon Tumor 2Classes, 95% in Leukemia 3Classes, 86% in Leukemia 4Classes and 100% in Brain Tumor 5Classes.

Keywords: Microarray data; Chi-square test; UpgCHI; method; feature selection.

1. Introduction

In the field of Microarray data classification, a diverse range of methods have been employed for the purpose of selecting pertinent attributes and ascertaining the ideal number of features from the extensive initial dataset [Prajapati and Gourisaria (2023)]. Microarray data has never been easy to categorize due to its huge dimensionality [Kim and Yoon (2022)]. These days, researchers can look at a single experiment done with over the genes 1000 because of the advancement of microarray data technology. Despite its small size, the numerical data within microarray datasets is of great value. It has a high computing cost and algorithmic instability because of its size. Several challenges in gene expression analysis are identified, such as a limited number of samples, insufficient validation, and an abundance of noisy and outlier-affected feature genes. Gene expression levels are typically assessed through microarray technology, with the resulting data applied to the diagnosis and treatment of diverse diseases.

A powerful tool for solving many biological problems, microarray technology has the potential to produce a wealth of information. The features in the raw dataset span from 6,000 to 60,000, however, training and testing samples are sometimes modest. (under 100 instances). Since this technique does not require all genes, feature selection becomes critical in order to extract useful genes. In the face of high-dimensional challenges, such as classification [Das and Behera (2019)] [Abdullah and Yap (2022)] [Sapri et al., (2022)] [Wan et al., (2022)] or clustering problems, researchers must rely on feature selection techniques [Hamla and Ghanem (2023)] to extract meaningful insights from their data and identify genes with significant biological connections [Das and Naik (2022)] [Behera et al., (2022)]. Because not every feature in the dataset can be used to make predictions, feature selection is necessary. The feature selection removes redundant and unnecessary characteristics from the dataset [Padhi and Chakravarty (2022)] [Naik et al., (2022)] [Pattanayak et al., (2022)] [Das et al., (2022)]. Various machine learning approaches [Silaich and Gupta (2022)] can be employed to assess the microarray

datasets. Thus, the ultimate objective of this paper is to feature selection for microarray data [Das and Naik (2019)] [Behera et al., (2019)].

Liang-jing Cai proposed the Var-CV-CHI algorithm, an enhancement to the CHI algorithm utilizing variance and coefficient of variation. This paper introduces a novel technique to address the flaws in the CHI algorithm. These experiments involve text classification in two languages (Chinese and English), using two different classifier algorithms (KNN and Naive Bayes), and working with two types of data distributions (balanced and unbalanced datasets). Importantly, it highlights that the results obtained after applying the proposed algorithm show significant improvements compared to the results achieved using the original CHI algorithm. The advantages of this paper are that the Classification effect is very excellent and compared to prior studies, this paper includes a more extensive set of experiments. This paper's limitation lies in its exclusive focus on the distribution of feature words, without taking into account their semantic information [Panigrahi and Moharana (2019)].

The authors addressed the challenge of selecting an appropriate technique in the absence of established standards for state-of-the-art findings within the scientific community [Cai and Liang-jing (2021)]. The primary objective of this study is to provide a powerful experimental comparison that evaluates the impact of the feature selection procedure when applied to different classification algorithms methods [Shu et al., (2021)]. In pursuit of this objective, they examined both ranking-based feature-selection procedures and cutting-edge strategies, facilitating a thorough comparison. The results obtained from various conventional microarray datasets, encompassing diverse attributes and patient counts, are presented comprehensively, shedding light on the outcomes achievable through these [Kai-bo et al., (2021)]. The authors introduced a χ^2 feature selection approach using Apache Spark, employing an algorithm using of Scikit-learn, and assessed its performance on the Databricks platform [Nassar and Mohamed (2019)] [Haidar et al., (2019)]. The aim of this paper is to diminish training time, enhance accuracy, and mitigate overfitting of the training data [Alaa et al., (2019)]. Future work involves researching and implementing advanced linear algebra, feature selection, and machine learning algorithms on the Apache Spark platform [Ahmed et al., (2019)] [Iskander et al., (2019)].

S. Bahassine presented an enhanced approach for Arabic text classification using Chi-square feature selection (ImpCHI) to improve classification performance [Bahassine and Said (2020)]. In the context of Arabic text classification, the quantity of pertinent terms for each class is well-established. Therefore, the Chi-square value is calculated for every term and across all classes [Abdellah et al., (2020)]. The ImpCHI algorithm aims to achieve a balanced selection of the number of attributes for each class [Mohammed et al., (2020)]. The forthcoming research in this paper aims to explore the prospect of extending the concept of attribute balance across classes to other feature selection algorithms [Mohamed et al., (2020)]. In an image analysis, Dixit, A. et al. (2020) [Dixit and Abhishek (2020)] utilized a blood cancer image dataset comprising 231 images. The DE-SVM classification demonstrates an impressive accuracy of 98.55%, surpassing the conventional SVM's accuracy of 86.96% [Ashish et al., (2020)]. Additionally, while the regular NB achieves an accuracy of 95.6%, the DE-NB attains a perfect accuracy score of 100% [Rohit et al., (2020)].

Previous research, however, was centered on various filter methods for classifying microarray data, typically limited to a small number of classes, often just two or three. In microarray data, there are various categories for diagnosis. It is desirable to achieve good performance in terms of accuracy, precision, recall, and F1-Score when classifying microarray data, particularly when dealing with multiple classes such as Colon Tumor 2Classes, Leukemia 3Classes and 4Classes and Brain Tumor 5Classes.

The problem definition of feature selection in multiclass microarray data to deal with the curse of dimensionality is a big problem. Small sample size, high dimensions and class imbalance are the most typical issues to overcome. In microarray data, the number of relevant genes belonging to each class doesn't know and occur class imbalance problem. Class imbalance can lower the credibility of classification accuracy. As the number of classes grows, the classification accuracy looks to rapidly deteriorate. One of the key components of microarray is the volume of quantitative information where number of features are very large with respect to samples. It can cause heavy computational cost in terms of space and time complexity and algorithmic instability.

The size of the enormous sample data is one of the biggest challenges in gene classification. A features selection method is used to eliminate pointless and unnecessary traits and choose the most distinctive features to get around this problem. The contributions in this paper are the UpgCHI algorithm, an upgrade to the Chi-square algorithm suitable for multiclass feature selection. To overcome the challenge of unknown gene relevance to each class, the original Chi-square is enhanced to UpgCHI. The study utilizes scalable platforms like Apache Spark to analyze microarray datasets and implements existing methodologies in this framework. The distributed frameworks such as MapReduce and Spark for the implementation of machine learning techniques, which are used to analyze the high-dimensional multiclass microarray datasets. Hence, the existing methodologies have been implemented on scalable platforms Apache Spark to analyze the microarray datasets.

2. Method

This paper introduces a new approach to selecting the feature and classifying microarray data into four and five different classes. This system uses an upgraded model by the original Chi-square test [Sikri and Alisha (2023)]. This method is implemented using the Apache Spark framework, and classifier performance is evaluated based on various performance metrics [Surjeet et al., (2023)].

2.1. Proposed Upgrade Chi-square Method (UpgCHI)

In this system, the suggested UpgCHI algorithm executes the chi-square values for all classes concurrently, which is different from the traditional Chi-square approach. The chi-square values are calculated for each class in microarray data where the gene expressions are unknown. The features with the greatest chi-square values are chosen from a set of attributes that belong to the same class. Up to the user-defined threshold, the count is achieved, and the top attributes for each class are chosen based on their ratio.

Algorithm of UpgCHI Test

Input:

n = user threshold count
class_number = number of classes
attributes_per_class = number of attributes for each class

Output:

Selected_features = []

Step1: Define the hypothesis:

-Null Hypothesis (H0): Two variables are independent.

-Alternate Hypothesis (H1): Two variables are not independent.

For i = 1 to class_number:

Alist[i] = []

For k = 1 to attributes_per_class:

Step 2: Build a contingency table for each attribute.

contingency_table = build_contingency_table(class_i_attribute_k)

Step 3: Calculate the expected value (Ei) for each instance.

Ei = expected_value(contingency_table) according to equation (1)

$$E_i = \left(T_{c_i} * \frac{T_{r_i}}{\sum_{k=1} T_{c_k}} \right) \quad (1)$$

Step 4: Calculate the chi-square value of each instance according to equation (2).

chi_square_value = calculate_chi-square(contingency_table, Ei)

$$Chi_Square = \frac{\sum_{i=1}^{no\ of\ instances} (O_i - E_i)^2}{E_i} \quad (2)$$

Step 5: Accept or Reject H0 df = degree_of_freedom(contingency_table) with alpha = 0.05 according to equation (3)

$$Df = N - 1 \quad (3)$$

if chi_square_value >= chi_square_value_from_distribution_table(df, alpha):

Reject the null hypothesis, accept the attribute, and save it.

Alist[i].append((class_i_attribute_k, chi_square_value))

Step 6: Select the top n attributes for class i.

Alist[i].sort(key=lambda x: x[1], reverse=True) # Sort by chi-square value.

Alist[i] = Alist[i][:n] # Select the top n attributes

Step 7: Select the final features by using SelectByRatio.

Selected_features = SelectByRatio(Alist, n)

```

# SelectByRatio function.
def SelectByRatio(Alist, n):
    final_selected_features = []
    ratios = []
    for i in range(1, class_number + 1):
        class_id = i
        num_attributes = len(Alist[i])
        total_instances = sum([len(a) for a in Alist])
        ratio = (total_instances / num_attributes) * 100
        ratios.append((class_id, ratio))
    # Sort the ratios in descending order
    ratios.sort(key=lambda x: x[1], reverse=True)
    z = 0
    while len(final_selected_features) <= n:
        final_selected_features.append(Alist[z][0])
        z += 1
    return final_selected_features
    
```

In the UpgCHI method, the microarray data where gene expressions are unknown. Thus, calculate the chi values for each class as a substitute. The UpgCHI algorithm aims to select the most significant attributes. To accomplish this, calculate the Chi-square value for every data point across all classes. From these values, retain the maximum Chi-square value (denoted as $\max(\text{Chi-square}(i, c))$ for each class c . These selected values then become attributes for their respective classes. Following that, attributes within the same class are arranged in order of their Chi-square values. The features with the most significant values in each class are selected to create a set of top attributes. The count of top attributes allocated to each class is determined by a specified ratio value. The calculation of this ratio value is as follows: it is obtained by taking the number of instances within each class, multiplying it by 100, and then dividing the result by the total number of instances.

Finally, this system selects features for each of the classes based on their ratio values until reach the user-specified threshold count. In this way, efficiently identify and retain the most relevant attributes for analysis. The unequal distribution of attributes across classes is a common issue in microarray data due to its unique characteristics. This uneven distribution can have a significant and detrimental impact on the classification process. In particular, the selected attributes are often not normalized properly in terms of their representativeness per class, further exacerbating the classification challenges. To address the problem caused by the absence of attributes in certain classes due to the chi-square approach, the adoption of the UpgCHI algorithm. This algorithm provides a solution to equalize the selection of attributes across classes, thus reducing the negative impacts of attribute dispersion and contributing to a more precise and efficient classification.

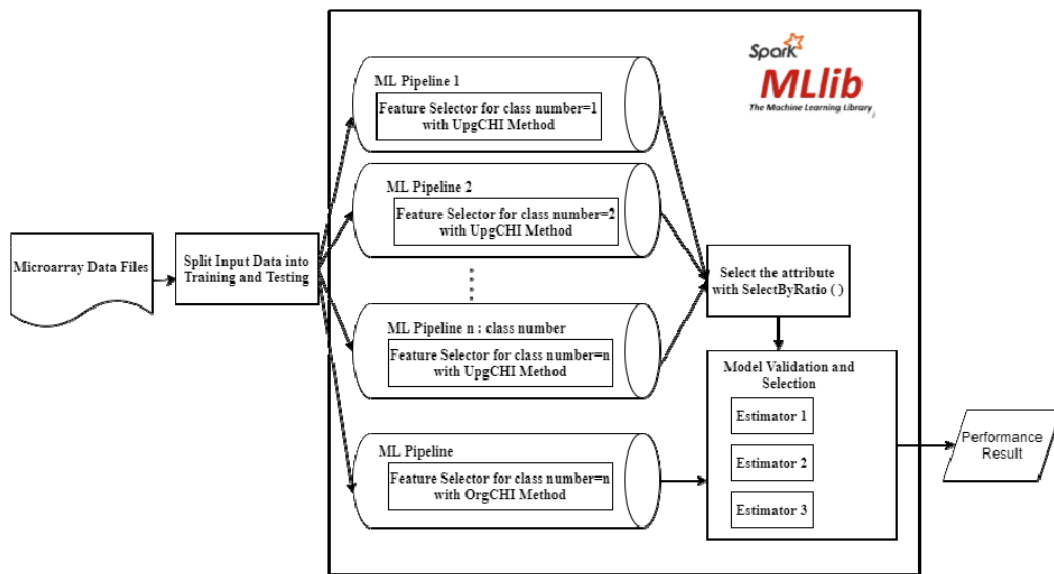


Fig. 1. Process Flow of Proposed System over Apache Spark

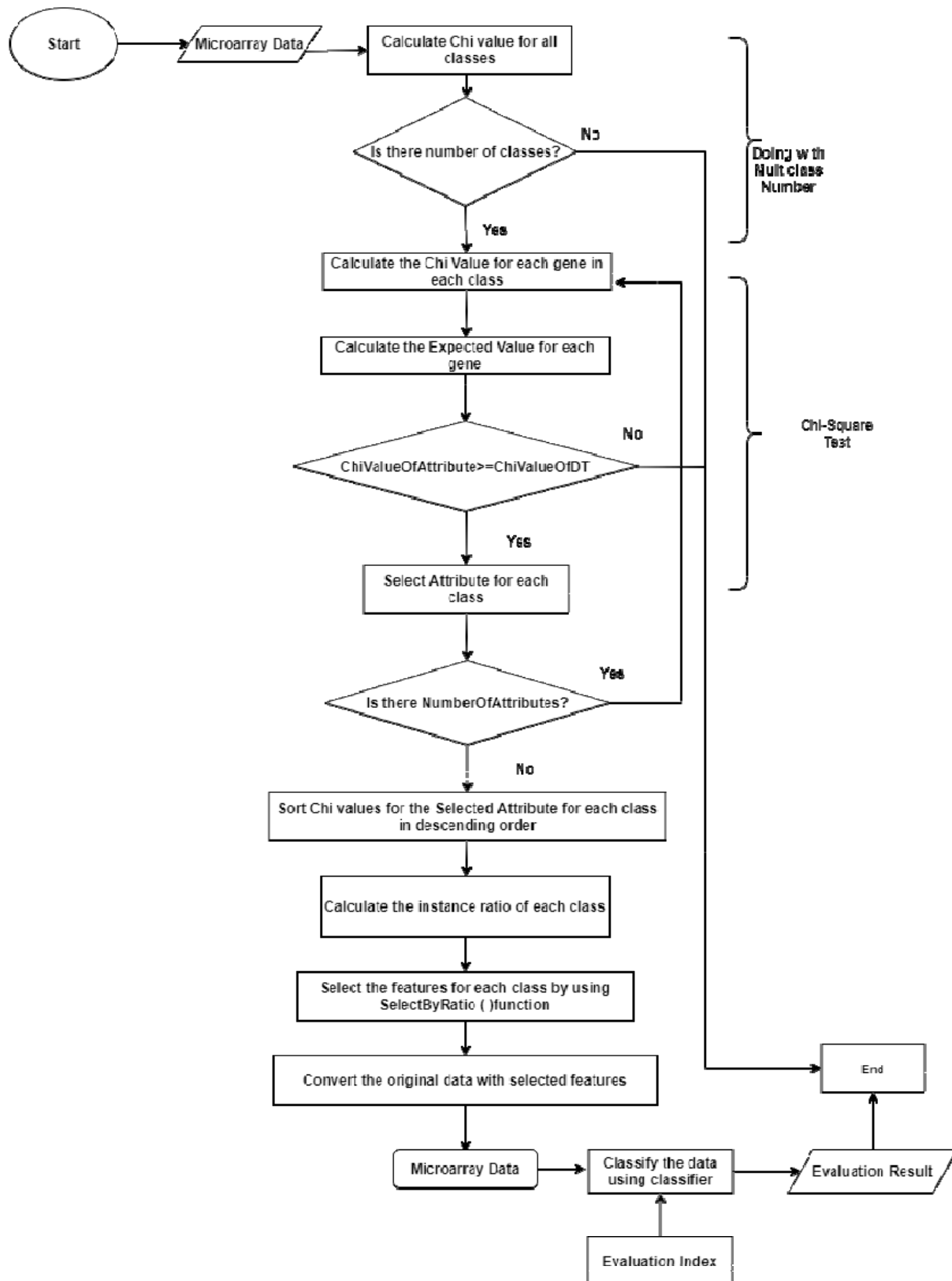


Fig. 2. Process Flow of Proposed System with UpgCHI Test Feature Selection

The proposed system utilizes filter-based chi-square feature selection methods. Specifically, this system applied the UpgCHI algorithm, taking into account the multiclass microarray data. UpgCHI demonstrates robustness in dealing with diverse data distributions, making it suitable for various scenarios. The proposed system utilizes filter-based chi-square feature selection methods. Specifically, this system applied the UpgCHI algorithm, taking into account the multiclass microarray data. UpgCHI demonstrates robustness in dealing with diverse data distributions, making it suitable for various scenarios. It offers ease of computation, provides detailed information from the tests, and exhibits flexibility in handling data from both two-group and multiple-group studies.

An Estimator abstracts the concept of a learning algorithm or any algorithm that fits or trains on data. For example, a learning algorithm such as Logistic Regression is an Estimator. Spark Mlib supports Logistic regression, Decision tree classifier, Random Forest classifier, Gradient-boosted tree classifier, Multilayer perceptron classifier, Linear Support Vector Machine, One-vs-Rest classifier (a.k.a. One-vs-All), Naive Bayes and Factorization machines classifier. In that system, system wants to use three classifiers Logistic Regression, Random Forest, and Naïve Bayes as Estimators. In Spark ML, logistic regression serves as a versatile tool for predicting outcomes. It enables the prediction of binary outcomes through binomial logistic regression or the prediction of multiclass outcomes via logistic regression in multinomial.

For classification involving multiple classes, Spark ML employs logistic (softmax) regression for multinomial. This method models the probabilities conditioned on the outcome classes ($k \in 1, 2, \dots, K$) with the use of softmax function. The optimization procedure entails the minimization of the weighted negative log-likelihood, employing a multinomial response model. Additionally, an elastic-net penalty is applied to the prevention of overfitting and to improve the model's generalization capabilities. Random Forests employ multiple decision trees to mitigate the risk of overfitting, making them a powerful ensemble learning technique. Similar to individual decision trees, random forests can be extended to handle multiclass classification, eliminating the need for feature scaling.

They are also adept at capturing complex relationships and interactions among features. Spark mllib offers support for random forests in classification for binary and multiple classes, as well as the tasks of regression, accommodating features in both continuous and categorical. By combining the predictions from each tree, Random Forests effectively reduce variance, leading to improved performance on test data. The aggregation of predictions from multiple trees helps to create a more robust and accurate model. Naive Bayes classifiers belong to a set of simple probabilistic multiclass classifiers by using Bayes' theorem, assuming a strong (naive) independence among every feature pair. Compared to other models, Naive Bayes classifiers demonstrate superior performance when trained with limited data. Naive Bayes algorithms are known for their quick processing, which can significantly save time during the training and prediction phases.

2.2. Datasets Description

Numerous websites, such as Kaggle, UCI, Mendeley Dataciteb12, the Global Health Observatory Data Repository, and many more, offer microarray datasets. You may easily store, share, access, and cite your data from anywhere with Mendeley Data, a safe cloud-based data repository. The following datasets were gathered by this approach from the Mendeley Data website: lymphoma, lung, breast, ovary, and CNS (central nervous system). There are enormous datasets available, with most of the features falling between 6000 and 60,000. Table 1 contains details about the datasets.

	Colon Tumor	Diffuse LargeB-Cell Lymphoma (DLBCL)	Leukemia	Leukemia	Brain Tumor
# Genes	2000	5469	7129	7129	5920
#Classes	2	2	3	4	5
#Instances	62	77	72	72	90

#Instances per Class	40/22	58/19	38/9/25	38/21/10/3	60/10/10/4/6
Class Names	B-cell differentiation, and proliferation	B lymphocytes or B cells	B-cell acute lymphoblastic leukemia, T-cell acute lymphoblastic leukemia and Acute myeloid leukemia	acute lymphoblastic leukemia (ALL), chronic lymphocytic leukemia (CLL), acute myeloid leukemia (AML) and chronic myeloid leukemia (CML).	Astrocytoma, Pilocytic Astrocytoma (grade I), Diffuse Astrocytoma (grade II), Anaplastic Astrocytoma (grade III), Glioblastoma Multiforme (grade IV),

Table 1. Experimental Microarray Dataset Details

3. Results and Discussion

This system evaluates the results using the proposed UpgCHI algorithm on microarray data. The experiments are performed using three classifiers: Random Forest, Logistic Regression and Naïve Bayes. The accuracy, F1-score, recall, and precision metrics are evaluated for various numbers of selected features. The results demonstrate that the UpgCHI algorithm outperforms the original Chi-square method in terms of classification accuracy. The experiments also highlight the effectiveness of the proposed algorithm for different numbers of features and classes.

In the process of computing the Upgrade Chi-Square test, it initially gathers the number of instances in each class. Next, this system calculates the chi-values for class 0, followed by class 1 and class 2, individually. Afterward, sort the chi-values in descending order within each class (class 0, class 1, and class 2). Lastly, the features are selected using the selectByRatio () function, adhering to the user-specified threshold count. This section covers the data processing and testing of the Logistic Regression, Random Forest, and Naïve Bayes algorithms. This system evaluates models using various metrics.

3.1. Colon Tumor 2Classes

In Colon Tumor 2Classes experiment, it works with 62 records and 2000 attributes. The results obtained Accuracy, F1-score, Recall, and Precision are presented for all three models. The experiment is conducted ten times, each data containing 50, 100, 150, 200, to 500 features. Each set of 50 features is repeatedly tested five times. Additionally, in Figure 3, throughout the test procedure, it was discovered that the nine times (450 features) had greater accuracy values for each evaluation metric. Figure 4's findings that Random Forest model has the higher accuracy of the 300 features. The findings of Naive Bayes model, in Figure 5, indicate that the selected feature 100 has greater accuracy. In table 2 shows the performance evaluation results and the average of it of the ten times of each algorithm. The results show that Naïve Bayes is better with average accuracy 89% compared to logistic regression with average of accuracy 88% and random forest with accuracy 82%. The classification accuracy results for three models (LR, RF and NB) in Table 3 and the results that the logistic Regression model has the higher accuracy with the selection of 450 features (95%).

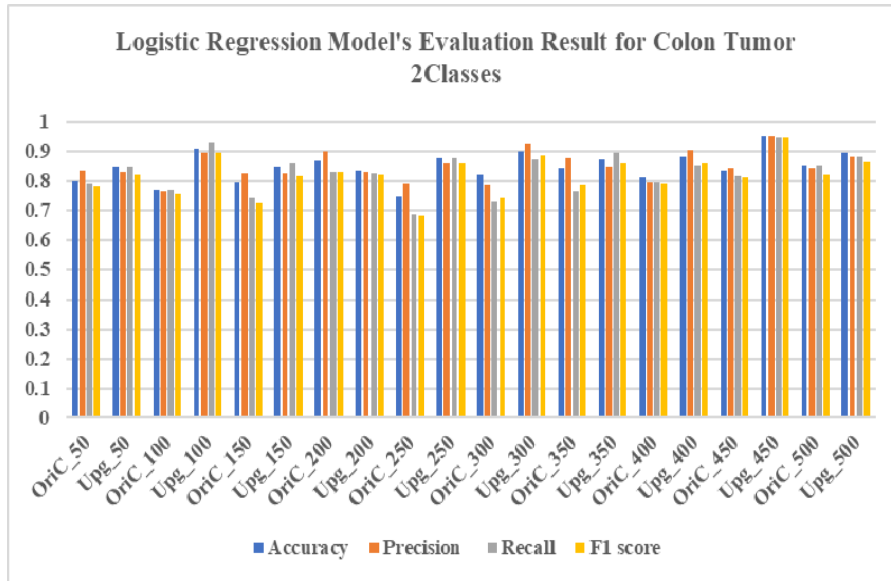


Fig. 3. Comparison of original Chi and UpgCHI with Logistic Regression Model using Colon Tumor 2Classes

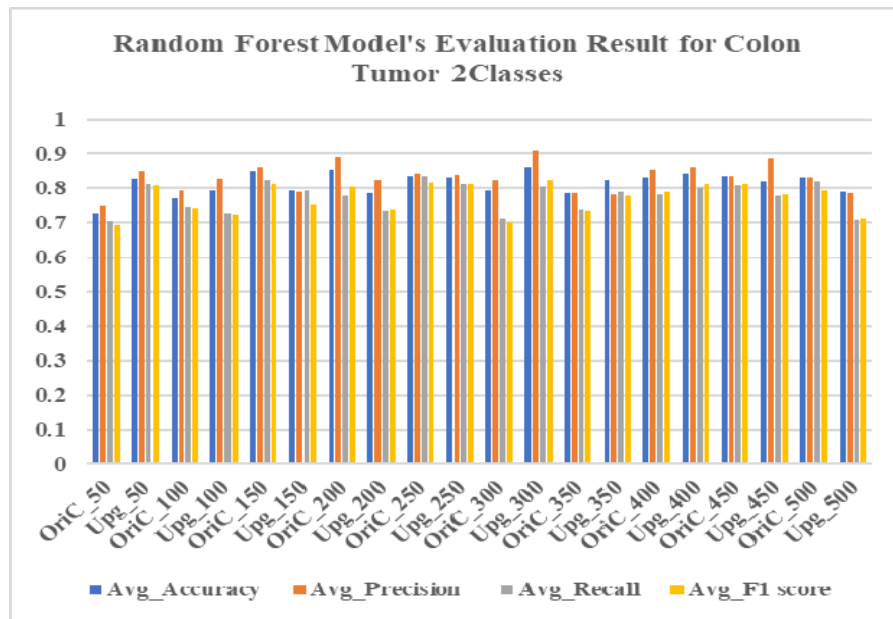


Fig. 4. Comparison of original Chi and UpgCHI with Random Forest Model using Colon Tumor 2Classes

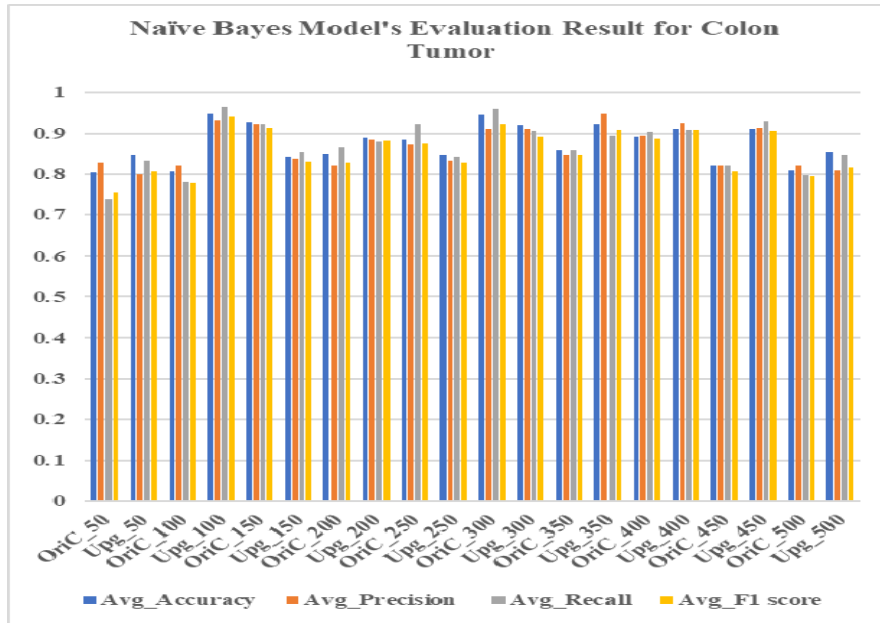


Fig. 5. Comparison of original Chi and UpgCHI with Naïve Bayes Model using Colon Tumor 2Classes

Evaluation Result	OrgCHI_LR	UpgCHI_LR	OrgCHI_RF	UpgCHI_RF	OrgCHI_NB	UpgCHI_NB
Accuracy	0.814	0.882	0.812	0.817	0.859	0.889
Precision	0.826	0.876	0.828	0.836	0.856	0.88
Recall	0.778	0.879	0.775	0.776	0.857	0.886
F1 score	0.773	0.863	0.77	0.775	0.841	0.871

Table 2. Comparison of Performance Analysis between Original Chi and UpgCHI using Colon Tumor 2Classes

Evaluation Result	Accuracy 50 Features	Accuracy 100 Features	Accuracy 150 Features	Accuracy 200 Features	Accuracy 250 Features	Accuracy 300 Features	Accuracy 350 Features	Accuracy 400 Features	Accuracy 450 Features	Accuracy 500 Features	Average
LR (Chi-Square)	0.710	0.824	0.866	0.807	0.909	0.896	0.867	0.913	0.887	0.918	0.860
LR (UpgCHI)	0.886	0.939	0.902	0.961	0.964	0.967	0.942	0.940	0.893	0.924	0.932
RF (Chi-square)	0.736	0.819	0.799	0.752	0.775	0.796	0.772	0.803	0.746	0.83	0.783
RF (UpgCHI)	0.857	0.796	0.770	0.850	0.882	0.754	0.799	0.793	0.849	0.853	0.820
NB (Chi-Square)	0.568	0.834	0.827	0.825	0.827	0.800	0.730	0.802	0.797	0.707	0.772
NB (UpgCHI)	0.677	0.820	0.765	0.753	0.721	0.872	0.715	0.748	0.847	0.837	0.776

Table 3. Comparison of Classification Accuracy over Logistic Regression (LR), Random Forest (RF) and Naïve Bayes (NB) between original Chi-square and UpgCHI using Colon Tumor 2Classes

3.2. DLBCL 2Classes

This system is evaluated the Accuracy, Precision, Recall and F1-score for DLBCL 2Classes (77 Records 5469 Attributes). The experiment is conducted ten times, each data containing 50, 100, 150, 200, to 500 features. Each set of 50 features is repeatedly tested five times. The average of it of the ten times of each algorithm and the results show in Table 4 that Logistic Regression is better with average accuracy 93% compared to Naïve Bayes with average of accuracy 77% and random forest with accuracy 82%. Table 5 shows the classification accuracy results for three models and the results show that the logistic Regression model has the higher accuracy with the selection of 300 features (96%).

Evaluation Result	OrgCHI_LR	UpgCHI_LR	OrgCHI_RF	UpgCHI_RF	OrgCHI_NB	UpgCHI_NB
Accuracy	0.860	0.932	0.783	0.820	0.772	0.776
Precision	0.846	0.907	0.728	0.761	0.740	0.751
Recall	0.811	0.921	0.689	0.751	0.771	0.792
F1 score	0.810	0.905	0.680	0.730	0.721	0.737

Table 4. Comparison of Performance Analysis between Original Chi and UpgCHI using DLBCL 2Classes

Evaluation Result	Accuracy 50 Features	Accuracy 100 Features	Accuracy 150 Features	Accuracy 200 Features	Accuracy 250 Features	Accuracy 300 Features	Accuracy 350 Features	Accuracy 400 Features	Accuracy 450 Features	Accuracy 500 Features	Average
LR (Chi-Square)	0.7106	0.824	0.866	0.807	0.909	0.896	0.867	0.913	0.887	0.918	0.8601
LR (UpgCHI)	0.8869	0.939	0.902	0.961	0.964	0.967	0.942	0.940	0.893	0.924	0.9323
RF (Chi-square)	0.7366	0.819	0.799	0.752	0.775	0.796	0.772	0.803	0.746	0.83	0.7833
RF (UpgCHI)	0.8576	0.796	0.770	0.850	0.882	0.754	0.799	0.793	0.849	0.853	0.8209
NB (Chi-Square)	0.5685	0.834	0.827	0.825	0.827	0.800	0.730	0.802	0.797	0.707	0.7721
NB (UpgCHI)	0.677	0.820	0.765	0.753	0.721	0.872	0.715	0.748	0.847	0.837	0.7761

Table 5. Comparison of Classification Accuracy over Logistic Regression (LR), Random Forest (RF) and Naïve Bayes (NB)between original Chi-square and UpgCHI using DLBCL 2Classes

3.3. Leukemia 3Classes

The outcomes of Accuracy, Precision, Recall, and F1-score for each algorithm using Leukemia 3Classes (72 Records and 7129 Attributes), as well as their average over ten trials, are displayed in Table 6. The findings indicate that Nave Bayes, with an average accuracy of 90%, outperforms Logistic Regression, which has an average accuracy of 85%, and Random Forest, which has an average accuracy of 78%. The classification accuracy results for the three models are shown in Table 7, and the findings indicate that the Nave Bayes model, which was chosen from 200 features, has the highest accuracy (95%).

Evaluation Result	OrgCHI_LR	UpgCHI_LR	OrgCHI_RF	UpgCHI_RF	OrgCHI_NB	UpgCHI_NB
Accuracy	0.776	0.859	0.655	0.789	0.866	0.901
Precision	0.789	0.837	0.600	0.714	0.863	0.903
Recall	0.753	0.800	0.578	0.701	0.847	0.915
F1 score	0.745	0.797	0.567	0.677	0.836	0.898

Table 6. Comparison of Performance Analysis between Original Chi and UpgCHI using Leukemia 3Classes

Evaluation Result	Accuracy 50 Features	Accuracy 100 Features	Accuracy 150 Features	Accuracy 200 Features	Accuracy 250 Features	Accuracy 300 Features	Accuracy 350 Features	Accuracy 400 Features	Accuracy 450 Features	Accuracy 500 Features	Average
LR (Chi-Square)	0.743	0.752	0.715	0.697	0.714	0.804	0.86	0.805	0.809	0.856	0.776
LR (UpgCHI)	0.856	0.801	0.768	0.892	0.849	0.877	0.898	0.873	0.897	0.884	0.859
RF (Chi-square)	0.596	0.641	0.527	0.599	0.695	0.705	0.699	0.702	0.624	0.763	0.655
RF (UpgCHI)	0.772	0.761	0.779	0.781	0.801	0.757	0.826	0.773	0.859	0.785	0.789
NB (Chi-Square)	0.836	0.859	0.783	0.774	0.81	0.905	0.889	0.887	0.941	0.972	0.866
NB (UpgCHI)	0.773	0.912	0.851	0.955	0.941	0.923	0.864	0.898	0.946	0.951	0.901

Table 7. Comparison of Classification Accuracy over Logistic Regression (LR), Random Forest (RF) and Naïve Bayes (NB) between original Chi-square and UpgCHI using Leukemia 2Classes

3.4. Leukemia 4Classes

In the Leukemia 4Classes experiment, it works with 72 records and 7129 attributes. The results obtained for Accuracy, F1-score, Recall, and Precision are presented for all three models. The experiment is conducted ten times, with data containing 50, 100, 150, 200, and 500 features each. Each set of 50 features is repeatedly tested five times. All three algorithms such as Logistic Regression (LR), Random Forest (RF), and Naïve Bayes (NB) demonstrate promising results in terms of accuracy, F-measure, recall, and precision. Table 8 shows the Accuracy, Precision, Recall, and F1-score results and the average of it ten times for each algorithm. The results show that Naïve Bayes is better with an average accuracy of 75% compared to Logistic Regression with an average of accuracy 69% and Random Forest with an accuracy of 60%.

Evaluation Result	OrgCHI_LR	UpgCHI_LR	OrgCHI_RF	UpgCHI_RF	OrgCHI_NB	UpgCHI_NB
Accuracy	0.6190	0.6925	0.5648	0.6066	0.7283	0.7542
Precision	0.4137	0.4897	0.3289	0.3960	0.5940	0.5705
Recall	0.4406	0.5449	0.3781	0.4699	0.6567	0.6420
F1-Score	0.4198	0.5139	0.35154	0.4272	0.6326	0.6296

Table 8. Comparison of Performance Analysis between Original Chi and UpgCHI using Leukemia 4Classes

Table 9 presents the classification accuracy results for three models, indicating that the Naive Bayes model achieves the highest accuracy when 500 features are selected (86%).

Evaluation Result	Accuracy 50 Features	Accuracy 100 Features	Accuracy 150 Features	Accuracy 200 Features	Accuracy 250 Features	Accuracy 300 Features	Accuracy 350 Features	Accuracy 400 Features	Accuracy 450 Features	Accuracy 500 Features	Average
LR (Chi-Square)	0.567	0.662	0.483	0.584	0.554	0.637	0.668	0.645	0.684	0.7	0.6184
LR (UpgCHI)	0.675	0.681	0.649	0.64	0.629	0.629	0.706	0.833	0.688	0.789	0.6919

RF (Chi-square)	0.43	0.576	0.543	0.571	0.554	0.528	0.617	0.602	0.593	0.63	0.5644
RF (UpgCHI)	0.622	0.607	0.592	0.649	0.521	0.565	0.659	0.634	0.562	0.65	0.6061
NB (Chi-Square)	0.616	0.687	0.686	0.705	0.743	0.802	0.82	0.849	0.814	0.757	0.7479
NB (UpgCHI)	0.679	0.607	0.787	0.7	0.664	0.755	0.802	0.85	0.835	0.86	0.7539

Table 9. Comparison of Classification Accuracy over Logistic Regression (LR), Random Forest (RF) and Naïve Bayes (NB)between original Chi-square and UpgCHI using Leukemia 4Classes

3.5. Brain Tumor 5Classes

Additionally, this system evaluates the Brain Tumor 5Classes (90 Records and 5920 Attributes) data and the result shows Accuracy, Precision, Recall, and F1 scores for each method, along with the average of those scores for the last ten iterations. The results show in Table 10 that Logistic Regression is better with an average accuracy of 90% compared to Random Forest with an average of accuracy 82% and Naïve Bayes with an accuracy of 88%.

Evaluation Result	OrgCHI_LR	UpgCHI_LR	OrgCHI_RF	UpgCHI_RF	OrgCHI_NB	UpgCHI_NB
Accuracy	0.866	0.902	0.799	0.820	0.868	0.882
Precision	0.685	0.750	0.581	0.601	0.717	0.767
Recall	0.675	0.782	0.564	0.601	0.714	0.796
F1-Score	0.667	0.756	0.551	0.579	0.702	0.763

Table 10. Comparison of Performance Analysis between Original Chi and UpgCHI using Brain Tumor 5Classes

The Logistic Regression model with the choice of 50 features has greater accuracy according to the classification accuracy findings for three models shown in Table 11. The performance assessment in the feature selection approach was evaluated during comparison stage by changing the quantity of the selected attributes. The results of the preceding figure, which measures classification accuracy, indicate that UpgCHI fared better than the original chi-square for the majority of the features. This system can infer from these findings that the UpgCHI algorithm helped to improve the categorization accuracy of Microarray data.

Evaluation Result	Accuracy 50 Features	Accuracy 100 Features	Accuracy 150 Features	Accuracy 200 Features	Accuracy 250 Features	Accuracy 300 Features	Accuracy 350 Features	Accuracy 400 Features	Accuracy 450 Features	Accuracy 500 Features	Average
LR (Chi-Square)	0.968	0.818	0.8	0.874	0.874	0.832	0.856	0.846	0.872	0.92	0.866
LR (UpgCHI)	1.008	0.866	0.906	0.856	0.92	0.916	0.886	0.894	0.862	0.906	0.902
RF (Chi-Square)	0.936	0.77	0.76	0.82	0.784	0.798	0.698	0.75	0.846	0.836	0.7998
RF (UpgCHI)	0.958	0.794	0.824	0.766	0.81	0.79	0.784	0.818	0.84	0.824	0.8208
NB (Chi-Square)	0.964	0.808	0.816	0.87	0.87	0.86	0.858	0.86	0.842	0.936	0.8684
NB (UpgCHI)	0.96	0.91	0.868	0.852	0.842	0.93	0.842	0.852	0.878	0.89	0.8824

Table 11. Comparison of Classification Accuracy over Logistic Regression (LR), Random Forest (RF) and Naïve Bayes (NB)between original Chi-square and UpgCHI using Brain Tumor 5Classes

The performance of the feature selection approach was evaluated during the comparison stage by changing the quantity of the selected attributes. The results of the preceding figure, which measures classification accuracy, indicate that UpgCHI fared better than the original chi-square for the majority of the features. This system can infer from these findings that the UpgCHI algorithm helped to improve the categorization accuracy of Microarray data.

4. Conclusion

Several classifiers utilizing the Spark framework have been created to categorize extensive microarray datasets. This system is aimed to improve the accuracy of microarray data classification using the proposed UpgCHI algorithm, based on the Chi-Square test. The use of Apache Spark as a scalable platform enhances the efficiency of the feature selection process. The experimental results demonstrate the superiority of the UpgCHI algorithm over the original Chi-square method in terms of classification accuracy. The proposed approach contributes to the field of feature selection and classification of multiclass microarray data. The outcomes suggest that the Logistic Regression classifier outperforms others in terms of accuracy, as confirmed by the UpgCHI test (Brain Tumor 5Classes) and the Naïve Bayes classifier is the best performance others in Leukemia 4Classes according to their selected features.

Acknowledgments

I want to extend my heartfelt gratitude and sincere thanks to my supervisor, my course leader, all of my class teachers, my family, and my friends at the University of Computer Studies in Yangon, Myanmar. Their unwavering kindness and faith in me have been a constant source of motivation. Their feedback, guidance, and wisdom have proven invaluable, and I owe much of my research progress to their invaluable guidance and insightful suggestions.



Conflicts of Interest

The authors have no conflicts of interest to declare.

References

- [1] Abdullah, M. N., Yap, B. W., Sapri, N. N. F. F., & Wan Yaacob, W. F., "Multi-class Classification for Breast Cancer with High Dimensional Microarray Data Using Machine Learning Classifier," In *Data Science and Emerging Technologies: Proceedings of DaSET 2022* (pp. 329-342). Singapore: Springer Nature Singapore.
- [2] Bahassine, Said, Abdellah Madani, Mohammed Al-Sarem, and Mohamed Kissi. "Feature selection using an improved Chi-square for Arabic text classification." *Journal of King Saud University-Computer and Information Sciences* 32, no. 2 (2020): 225-231.
- [3] Das H, Naik B, Behera HS, "A Jaya algorithm-based wrapper method for optimal feature selection in supervised classification," *J King Saud Uni-Comp Inform Sci.* 2022;34(6):3851–63.
- [4] Das, H., Naik, B., & Behera, H. S., "An experimental analysis of machine learning classification algorithms on biomedical data," In *Proceedings of the 2nd International Conference on Communication, Devices and Computing: ICCDC 2019* (pp. 525-539). Springer Singapore.
- [5] Das, H., Naik, B., & Behera, H. S., "Disease classification using linguistic neuro-fuzzy model," In *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019* (pp. 45-53). Springer Singapore.
- [4] Dixit, Abhishek, Ashish Mani, and Rohit Bansal. "Feature selection for text and image data using differential evolution with SVM and naïve Bayes classifiers." *Engineering Journal* 24, no. 5 (2020): 161-172.
- [5] Hamla H, Ghanem K, "A Comparative Study of Filter Feature Selection Methods on Microarray Data," In: *12th International Conference on Information Systems and Advanced Technologies "ICISAT 2022"* Intelligent Information, Data Science and Decision Support System. Cham: Springer International Publishing; 2023. p. 186–201.
- [6] Kim J, Yoon Y, Park HJ, Kim YH, "Comparative study of classification algorithms for various DNA microarray data," *Genes.* 2022;13(3):494.
- [7] Nassar, Mohamed, Haidar Safa, Alaa Al Mutawa, Ahmed Helal, and Iskander Gaba. "Chi squared feature selection over Apache Spark." In *Proceedings of the 23rd International Database Applications & Engineering Symposium*, pp. 1-5. 2019.
- [8] Padhi BK, Chakravarty S, Naik B, Pattanayak RM, Das H. RHOF, "Feature selection using the rock hyrax swarm optimization algorithm for credit card fraud detection system," *Sensors.* 2022;22(23):9321.
- [9] Panigrahi, K. P., Das, H., Sahoo, A. K., & Moharana, S. C., "Maize leaf disease detection and classification using machine learning algorithms," In *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019* (pp. 659-669). Springer Singapore.
- [10] Prajapati, S., Das, H. & Gourisaria, M.K, "Feature selection using differential evolution for microarray data classification," *Discover Internet Things* 3, 12 (2023). <https://doi.org/10.1007/s43926-023-00042-5>
- [11] Sikri, Alisha, N. P. Singh, and Surjeet Dalal. "Chi-Square Method of Feature Selection: Impact of Pre-Processing of Data." *International Journal of Intelligent Systems and Applications in Engineering* 11, no. 3s (2023): 241-248.
- [12] Silaich, S., & Gupta, S., "Feature Selection in High Dimensional Data: A Review," In *Third Congress on Intelligent Systems: Proceedings of CIS 2022, Volume 1* (pp. 703-717). Singapore: Springer Nature Singapore.

Authors Profile

	<p>Lwin May Thant hold a Master of Computer Science degree from the University of Computer Studies, Maubin in 2010. She also received her B.C.Sc. and B.C.Sc. (Hons:) from the University of Computer Studies, Maubin in 2007 and 2008, respectively. She is currently a Lecturer at the Faculty of Computer Science at the University of Computer Studies, Maubin. She is currently a research candidate at the University of Computer Studies, Yangon. Her research includes machine learning, data mining, data analysis and bioinformatics. She can be contacted at email: lwinmaythant@ucsy.edu.mm.</p>
	<p>Tin Zar Thaw is currently a professor in the Faculty of Computer Science, University of Computer Studies, Yangon. She received her Master's degree in Computer Science and the PhD degree in information Technology, University of Computer Studies, Mandalay, in 2004 and 2013 respectively. Her research interests include Web Engineering, Deep Learning, Machine Learning, Cloud Computing, Big Data Analytics.</p>